A ROBUST PRECONDITIONING TECHNIQUE FOR LARGE, SPARSE LEAST SQUARES PROBLEMS

Miroslav Tůma
Institute of Computer Science,
Czech Academy of Sciences, Prague, and
TU Liberec, Czech Republic

Joint work with

Michele Benzi
Dept. of Mathematics and Computer Science
Emory University,
Atlanta, GA 30322, USA

**XV. Householder Symposium,
Peebles, Scotland, June 17–21, 2002,**

OUTLINE

1. The Problem and the Method: Large Sparse Linear Least Squares and Preconditioned CGNR
2. Previous Work
   - LU-based Preconditioning
   - Incomplete QR Preconditioning
   - Incomplete Cholesky Preconditioning
   - New Approach: Incomplete Decomposition via A-Orthogonalization
3. A-Orthogonalization – Basic Features
4. A-Orthogonalization – Incompleteness and Preconditioning
5. A-Orthogonalization – Ways to Direct Decompositions
6. RIF and Linear Least Squares
7. Numerical Experiments
8. Conclusions

1) The Problem and the Method

Solving linear least-squares problem

$$\min_x ||Bx - b||_2 \tag{1}$$

- B is $m \times n$ with $m \geq n$,
- B is large and sparse
- $\text{rank}(B) = n$

Our method of choice:
the **preconditioned** conjugate gradient
method on the normal equations (CGNR)

Needed: a good **general purpose** (algebraic) and **robust** preconditioner:

$$\min_y ||BM^{-1}y - b||_2, \quad Mx = y \tag{2}$$

Desired properties of the preconditioner (see Björck, 1996):

- BM^{-1} better conditioned than B
- cheap application of M^{-1} and M^{-T}

Interested in preconditioners based on
incomplete factorizations

satisfying (hoping to satisfy):

- $A = B^T B$ is not explicitly computed
  - some matrices provide very dense $A = B^T B$ (see Björck, 1996)
  - a lot of preconditioner computations can be based on having A only implicitly but we usually pay by the speed. **Need of fast implicit decompositions**
  - the approach using explicit computations of A is often more efficient in practice

- factorization should be breakdown-free
  - IC breakdown-free factorizations of B^TB
  - nonsymmetric orderings and/or pivoting for decomposition of B in LU-based methods
- memory efficient
  - it is fine to pay with memory for the speed within reasonable bounds
  - sometimes difficult to avoid costly and unreliable preconditioners

2) Previous Work

- **1. LU-based methods**
  - Laüchli, 1961

$$B = \begin{pmatrix} M \\ B_2 \end{pmatrix}$$

Explicit application of the preconditioner with the normal equations as:

$$(BM^{-1})^T(BM^{-1}) = I + C^TC.$$

Gauss-Jordan with complete pivoting.
No sparsity considerations.

  - Saunders, 1979

Partial pivoting, preconditioned by U^{-1}

  - Björck, Yuan, 1999

Using Gilbert-Peierls partial pivoting code.

Notes on LU-based preconditioning for CGNR:

- Might be difficult to find the decomposition of B cheaply when employing a direct solver
- incomplete decompositions of B?

Proposals and/or future work:

- in any case, seem to be **dependent on a nonsymmetric reordering** of the matrix; it strongly enhances quality of the choice of the first rows
- the strategy of Olschowka, Neumaier, 1996; Koster, Duff, 1998, and their MC64 code can be used but **should be modified** in order to take into account sparsity of individual rows
- two levels of preconditioners: LU / IC ?

- **2. Incomplete QR and inverse QR methods:**

  $B \approx QR$ with $Q^T Q \approx I$, R upper triangular

  $$\Longrightarrow B^T B \approx R^T R.$$

  - Jennings, Ajiz, 1984 (Gram-Schmidt, Givens)
  - Zlatev, Nielsen, 1988 (Givens)
  - Saad, 1988 (ILQ, Gram-Schmidt)
  - James, 1990 (Givens)
  - Wang, Gallivan, Bramley, 1997 (CIMGS)
  - Saad, Sosonkina, 2001 (shifted ILQ)
  - Bai, Duff, Wathen, 2001 (Givens)
  - Papadopoulos, Duff, Wathen, 2002 (Givens)
  - Benzi, T., 1995 (inverse Gram-Schmidt)

Notes on QR-based preconditioning for CGNR:

- Which scheme? Both Gram-Schmidt and Givens can be naturally safeguarded against breakdowns (Ajiz, Jennings, 1984)
- CIMGS: reformulated incomplete Gram-Schmidt
- preconditioners can be very reliable (in some applications)
- reliability comes sometimes at the very high cost
- crucial role of dropping rules
(cf. Papadopoulos, Duff, Wathen, 2002)
- **missing a guiding structure like in many LL^T/LU decompositions** – dense preconditioners more often useless than in other incomplete decompositions
- incomplete Householder: an overlooked choice to be explored

- **3. Incomplete Cholesky decomposition of $A = B^T B$:**

  The most traditional approach

  - problem of breakdowns: best safeguards are memory "hogs"
  - $\rightarrow$ collapse of data structure paradigms
  - should be computed implicitly

- **4. New approach: decomposition via A-orthogonalization:**

  Motivation:

  - breakdown-free
  - fill-in not only local like in direct decompositions (factorized matrix inverse behind the construction)
  - nevertheless: even more difficult to stay sparse,

3) A-Orthogonalization – Basic Features

For an SPD matrix A find an upper triangular Z and diagonal matrix D such that

$$Z^T A Z = D \to A^{-1} = Z D^{-1} Z^T \tag{3}$$

- The algorithm: Conjugate Gram-Schmidt: GS with a different inner product: $(x, y)_A$
- Origins of the A-orthogonalization for solving linear systems in more papers in 40's
- A more detailed treatment of the A-orthogonalization: in the first Wilkinson paper (with Fox and Huskey, 1948)

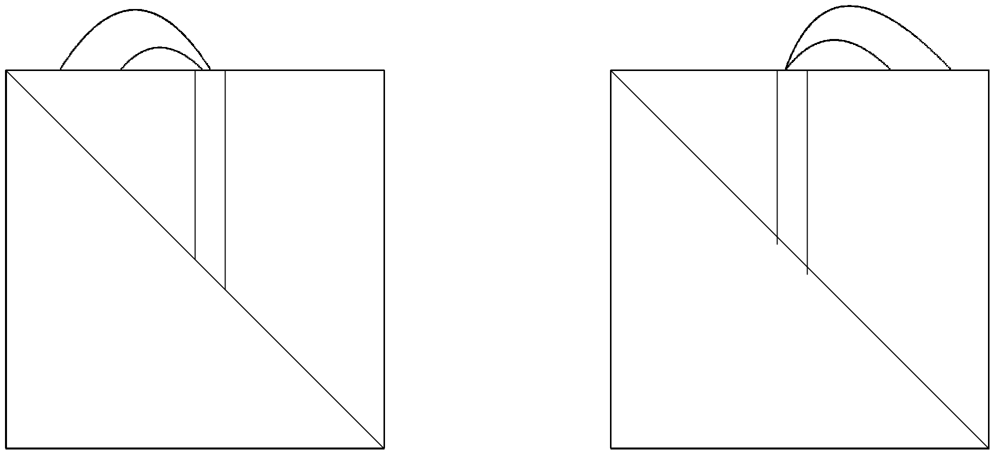
First systematic treatment of the technique: Hestenes, Stiefel, 1952 (conjugate direction methods).

Specific features, Hestenes-Stiefel, 1952:

- only dense matrices considered; used as a direct method
- not covering computational issues
- *classical* versus *modified* algorithm
- two basic ways to compute diagonal entries

$\boxed{\text{Algorithm H-S I.}}$

$$z_i = e_i - \sum_{k=1}^{i-1} z_k \frac{e_i^T A z_k}{z_k^T A z_k}, i = 1, \ldots, n; Z = [z_1, \ldots, z_n]$$



- left-looking
- *stabilized* diagonal entries (in exact arithmetic $e_i^T A z_k \equiv z_i^T A z_k, i \leq k$)

$\boxed{\text{Algorithm H-S II.}}$

$$z_i^{(j+1)} = z_i^{(j)} - z_i^{(j)} \frac{e_j^T A z_i^{(j)}}{e_j^T A z_j}$$

$$j = 1, \ldots, n; i = j+1, \ldots, n; Z = [z_1, \ldots, z_n]$$

- right-looking
- *non-stabilized* diagonal entries

A-orthogonalization in historical perspective

- Fox, Huskey, Wilkinson, 1948: H-S I.

- Escalator method by Morris, 1946: a variation of H-S I. (non-stabilized computation of D),

- Vector method by Purcell, 1952: basically H-S II.

- Approximate inverse by bordering (Saad, 1996) is *equivalent* to H-S I. (Benzi, T. (2002))

- Bridson, Tang, 1998 – (nonsymmetric) algorithms equivalent to H-S I.

4) A-Orthogonalization – Incompleteness and Preconditioning

Incomplete decompositions

1. Possibility of *breakdowns*
2. *Classical* versus *modified*

Possibility of breakdowns

- AINV: modified incomplete A-orthogonalization; exists for M-matrices and H-matrices (Benzi, Meyer, T., 1996)
- Possibility of breakdown in A-orthogonalization for non-H matrices
- Possibly poor approximate inverses for these matrices

Possibility of breakdowns (continued)

- SAINV: stabilized computation in A (Benzi, Cullum, T. (2000); Kharchenko et al. (2001))

$$z_i = e_i - \sum_{k=1}^{i-1} z_k \frac{e_i^T A z_k}{z_k^T A z_k}, i = 1, \ldots, n$$

$$\downarrow$$

$$z_i = e_i - \sum_{k=1}^{i-1} z_k \frac{z_i^T A z_k}{z_k^T A z_k}, i = 1, \ldots, n$$

- Stabilization of diagonal entries is enough only for the breakdown-free property in SPD case but not for having a good preconditioner.
- Difficult problems (structural mechanics of thin shells, diffusion equations on complicated geometries and skewed meshes): SAINV clearly outperforms DCR, H-S I.

- For easy problems typically similar behavior of AINV and SAINV
- Complexity of SAINV typically not much larger than this of AINV $\Longrightarrow$ sparsity, implementation issues

Classical versus modified

- *Modified* algorithms might be better than the *classical* ones. Why?
- Incomplete *modified* algorithms: AINV (Benzi, Meyer, T., 1996)
- Incomplete *classical* algorithms? AIB (Saad, 1996). Breakdown free for SPD matrices.
- *Limited* experiments with a *specific* implementation of AIB show that is less powerful than AINV
- If this is so, a question: is there a "modified" version of the update formulas for Sherman-Morrison formula based preconditioners (Bru, Cerdán, Marín, Mas, 2002)?

Other possible stabilization attempts:

- Pivoting
- Look-ahead
- DCR
- Block algorithms

5) A-Orthogonalization – Ways to Direct Decompositions

First approach

AIB (classical Conjugate GS approximate inverse decomposition, see Saad, 1996)

$$\begin{pmatrix} Z^T & \\ -y^T & 1 \end{pmatrix} \begin{pmatrix} A & v \\ v^T & \alpha \end{pmatrix} \begin{pmatrix} Z & -y \\ & 1 \end{pmatrix} = \begin{pmatrix} D & \\ & \delta \end{pmatrix}$$

$y = ZD^{-1}Z^T v$ with dropping

$+$ stabilization of diagonal entries

$$\delta = \alpha - y^T v \to \delta = \alpha - 2y^T v + y^T A y$$

$$\downarrow$$

provides a breakdown-free procedure since

$$\delta = \begin{pmatrix} -y^T & 1 \end{pmatrix} \begin{pmatrix} A & v \\ v^T & \alpha \end{pmatrix} \begin{pmatrix} -y \\ 1 \end{pmatrix}$$

an idea: direct decomposition by bordering based on:

$$\begin{pmatrix} A & v \\ v^T & \alpha \end{pmatrix} = \begin{pmatrix} L & \\ y^T & 1 \end{pmatrix} \begin{pmatrix} D & \\ & \delta \end{pmatrix} \begin{pmatrix} L^T & y \\ & 1 \end{pmatrix}$$

$y = D^{-1} L^{-1} v$ with dropping

$+$ stabilization of diagonal entries

$$\delta = \alpha - y^T D y \to \delta = \alpha - 2 y^T D y + y^T L^{-1} A L^{-T} y$$

$\downarrow$

stabilized ILUS (Chow, Saad, 1995; see also Bollhöfer, Saad, 2001) (breakdown-free procedure since

$$\delta = \begin{pmatrix} y^T L^{-1} & -1 \end{pmatrix} \begin{pmatrix} A & v \\ v^T & \alpha \end{pmatrix} \begin{pmatrix} L^{-T} y \\ -1 \end{pmatrix}$$

sometimes better than ILUS

analogical “stabilization” approach as making AINV $\rightarrow$ SAINV:

stabilized incomplete Cholesky by bordering (ICB) based again on

$$\begin{pmatrix} A & v \\ v^T & \alpha \end{pmatrix} = \begin{pmatrix} L & \\ y^T & 1 \end{pmatrix} \begin{pmatrix} D & \\ & \delta \end{pmatrix} \begin{pmatrix} L^T & y \\ & 1 \end{pmatrix}$$

$$y = L^{-1}AL^{-T}L^{-1}v$$

$$\delta = \alpha - 2y^TDy + y^TL^{-1}AL^{-T}y$$

$$\downarrow$$

DISASTER

Second approach

LDL^T decomposition can be a side-product of the A-orthogonalization; Benzi, T. (2001)

$$Z^T A Z = D, \ A = LDL^T \Longrightarrow L = AZD^{-1}$$

entries of L = multipliers in orthogonalization

$$\frac{z_j^T A z_i}{z_j^T A z_j}$$

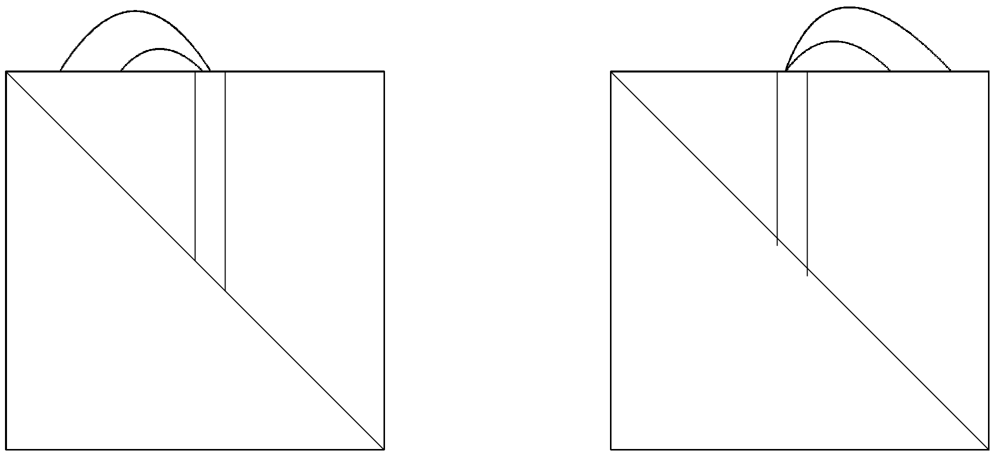
(stabilized case)
(cf. Hestenes, Stiefel, 1952 for relation of A-orthogonalization and Gaussian elimination)

- LDL^T factorization as a side-product of A-orthogonalization

- incompleteness: stabilized incomplete A-orthogonalization (SAINV) determines incomplete decomposition (RIF)

- breakdown-free for SPD systems

- upper triangular Z by columns $\times$ upper triangular L^T by rows



- especially for lower density preconditioners better than other stabilized IC

6) RIF and Linear Least Squares

Find Z such that

$$Z^T B^T B Z = D \to (B^T B)^{-1} = Z D^{-1} Z^T \quad (4)$$

RIF approach

1) SAINV: $z_i = e_i - \sum_{k=1}^{i-1} z_k \frac{\langle Bz_k, Bz_i \rangle}{\langle Bz_k, Bz_k \rangle}$,

$$i = 1, \ldots, n; Z = [z_1, \ldots, z_n]$$

2) $Z \to L$ using multipliers in SAINV

7) Numerical experiments

Method for comparison presented in this talk

- (Value based) incomplete Cholesky for normal equations (standard used in comparisons)

Other approaches (tested so far):

- Incomplete LU preconditioning with agressive choice of diagonal entries (MC64; Duff, Koster, 1999) and subsequent partial pivoting

- AINV for normal equations

- Incomplete QR (some types)

Example matrices in experiments

- Well-known matrices from HB collection
- Animal breeding matrices from CERFACS (`ftp://ftp.cerfacs.fr/pub/algo`)
- Original ordering, no scaling, no prefiltration, no postfiltration

Matrix	m	n	nnz
WELL1033	1033	320	4732
ILLC1033	1033	320	4732
WELL1850	1850	712	8758
ILLC1850	1850	712	8758
SMALL	3140	1988	8510
MEDIUM	9397	6119	25013
LARGE	28524	17264	75018
VERYL	174193	105882	463303

Experiments – comparison of two “best” approaches (well-known matrices)

Matrix	RIF			
	P(size)	t(P)	Its	t(It)
WELL1033	911	0.03	72	0.04
ILLC1033	825	0.03	256	0.13
WELL1850	2835	0.05	89	0.08
ILLC1850	2904	0.04	248	0.23

Matrix	IC by value			
	P(size)	t(P)	Its	t(It)
WELL1033	866	0.005	108	0.05
ILLC1033	816	0.005	286	0.14
WELL1850	2595	0.01	182	0.17
ILLC1850	2691	0.01	774	0.72

- Marginal differences (some stuff dominated by setup costs)
- Much larger sensitivity of IC with respect to dropping
- Non-uniform behavior of IC convergence: fill/**its** for ILLC1850: 1868/**1013**; 2904/**774**; 3229/**2515**; 3461/**5393**

Experiments – comparison of two "best" approaches (larger matrices)

Matrix	RIF			
	P(size)	t(P)	Its	t(It)
SMALL	**12199**	**0.22**	**32**	**0.05**
MEDIUM	**28420**	**0.20**	**53**	**0.31**
	37248	0.70	34	0.21
LARGE	**76094**	**0.54**	**63**	**0.96**
	82287	0.70	58	0.91
	107007	3.62	39	0.70
VERYL	**426830**	**5.18**	**199**	**26.0**
	272207	3.10	225	34.9
	249327	2.17	206	31.4

Matrix	IC by value			
	P(size)	t(P)	Its	t(It)
SMALL	**17377**	**0.05**	**33**	**0.06**
	13443	0.03	134	0.24
MEDIUM	**55950**	**0.14**	**246**	**1.50**
	47714	—	0.11	—
LARGE	**224430**	**1.38**	**38**	**1.00**
	180309	0.85	—	—
	163157	0.69	—	—
VERYL	1185773	7.71	127	31.9
	962079	**5.72**	**135**	**29.2**
	730622	4.02	507	104.

- IC tends to have much larger fill-in in factors
- RIF (breakdown-free) clearly a reasonable choice
- Time to compute the RIF preconditioner increases for larger fill-in more quickly than for IC

Notes on other approaches and other notes on experiments

- AINV preconditioning
  - Behaves similarly to RIF. Switch to RIF (combined with postfiltration) typically decreases fill in factors; but removes explicitness.
- Incomplete LU preconditioning
  - Incomplete LU is strongly dependent on static preordering to maximize module of product of diagonal entries of the matrix

- The preconditioner is sparser but less efficient

- Two levels should be done.

- Incomplete QR preconditioning

  - We restricted to IMGS type

  - Large memory requirements for the MGS type of incomplete QR:

    * Large dynamic data structures (for standard implementations)

    * Large dynamic or static data structures for Tismenetsky, 1991 approach (for left-looking versions see Wang et al., 1997 and Kaporin ,1999 and later

Another matrices:
traj27bt and **traj33bt**
from Boeing

- behave similarly, we will describe **traj27bt**
- n/nonz: 7145 / 75703
- very dense $A = B^T B$: 52051025 nonzeros

	IC by value			
Method	P(size)	t(P)	Its	t(It)
IC/CGNR	**14288**	**18.9**	**243**	**1.57**
RIF/CGNR	**16479**	**47.3**	**255**	**1.67**
ILU/CGNR	**14665**	**12.5**	**2025**	**15.2**
IMGS/CGNR	**15555**	**13.8**	**757**	**5.15**

8) Conclusions

- Hestenes, Stiefel 1952 paper: motivating source even now
- A lot of work to be done for IQR and LU-based preconditioning of CGNR
- Natural use of A-orthogonalization for solving least-squares problems
- RIF (breakdown-free preconditioner) for CGNR might be a reasonable method of choice for solving linear least squares

Thank you!

(missing picture of Nessie)